HW10: Application exercises

Ellen Hsieh

```
library(tidyverse)
library(tidymodels)
library(Rtsne)
library(scales)
library(factoextra)
library(cluster)
library(iml)
library(glmnet)
library(caret)
library(earth)
library(randomForest)
library(klaR)
library(tictoc)
library(ggfortify)
library(dplyr)
library (plyr)

set.seed(124)
```

Dimension reduction

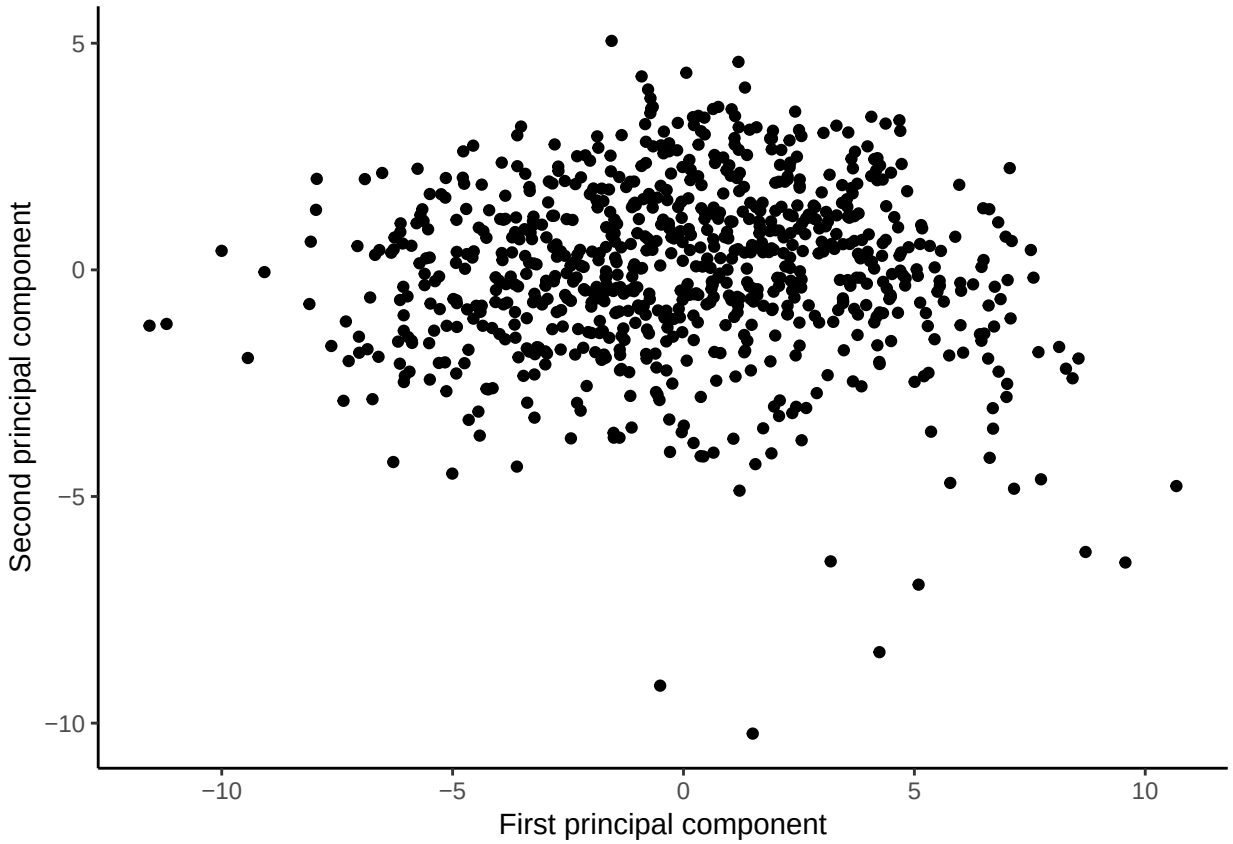
```
wiki_data <- read_csv("data/wiki.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer()
## )
```

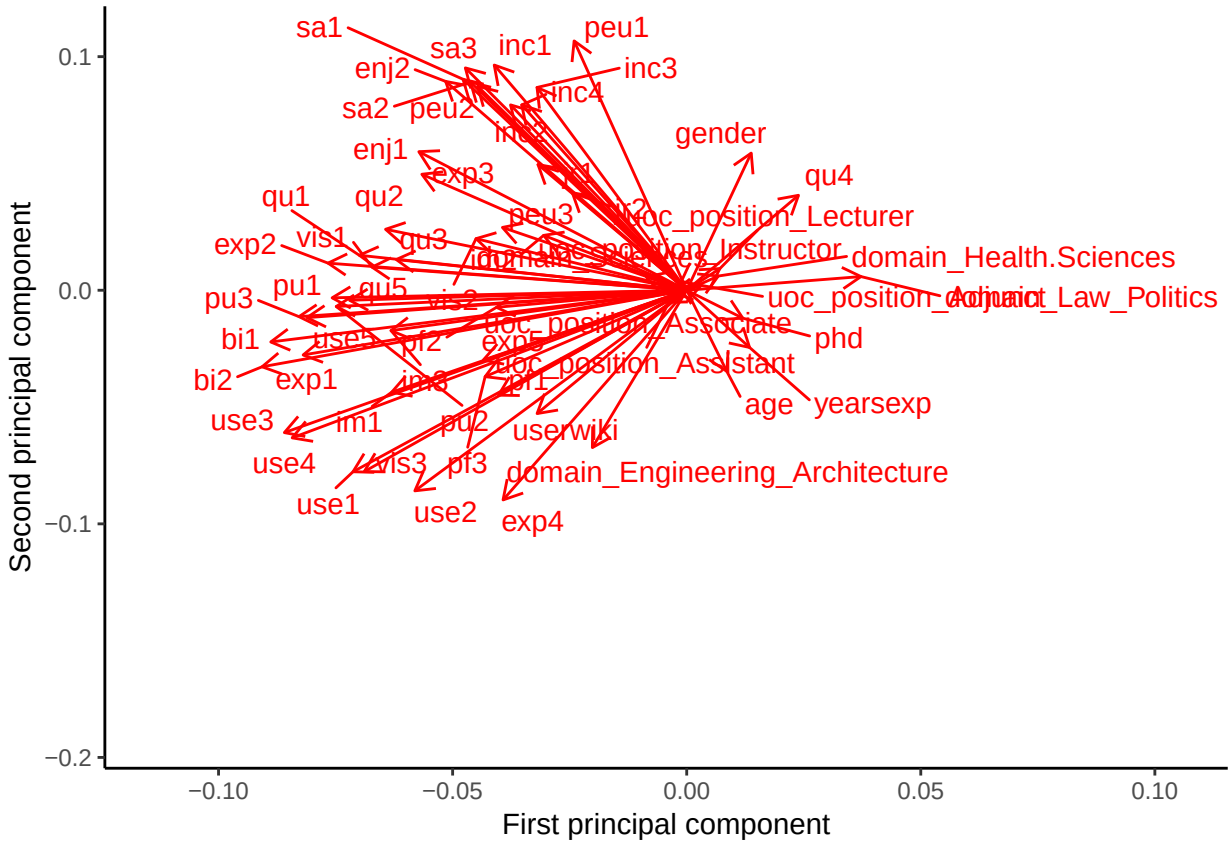
```
## See spec(...) for full column specifications.
```

1.

```
wiki_pca <- prcomp(wiki_data, scale= TRUE)

pc1 <- wiki_pca$x[,1]
pc2 <- wiki_pca$x[,2]
pca_data <- tibble(pc1, pc2)

ggplot(data = pca_data, aes(x = pc1, y = pc2)) +
  geom_point() + theme_classic() +
  labs(x = "First principal component",
       y = "Second principal component")
```

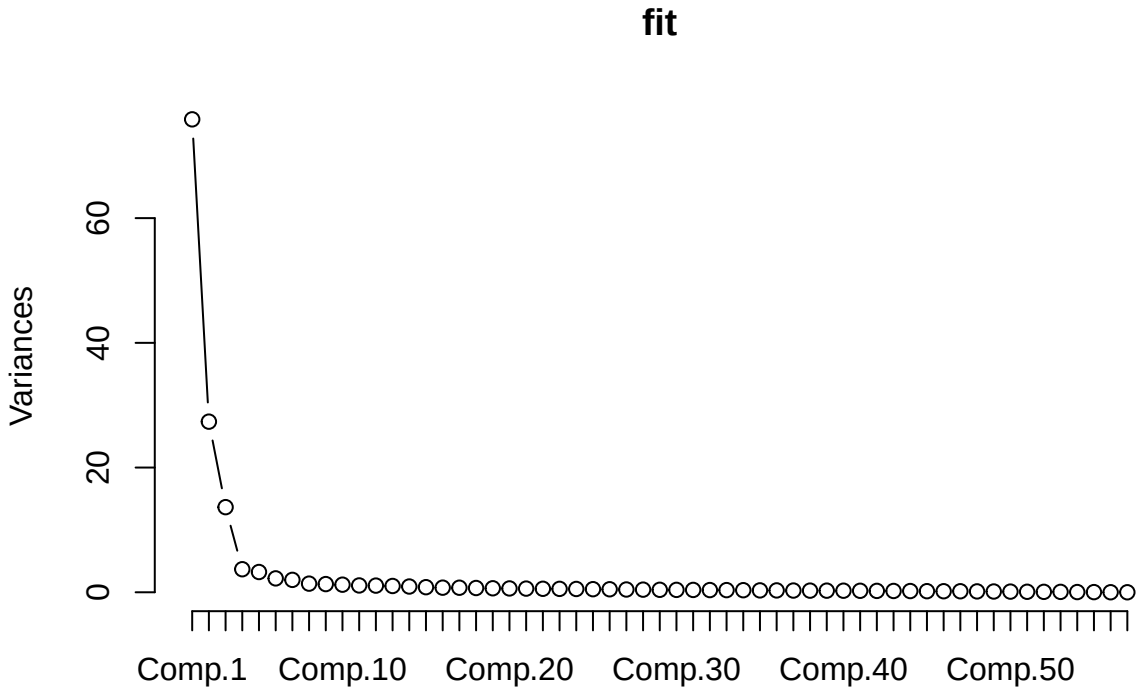



```
autoplot(wiki_pca, alpha = 0, loadings = TRUE,
         loadings.label = TRUE, loadings.label.repel = TRUE) + theme_classic() +
         labs(x = "First principal component",
              y = "Second principal component")
```

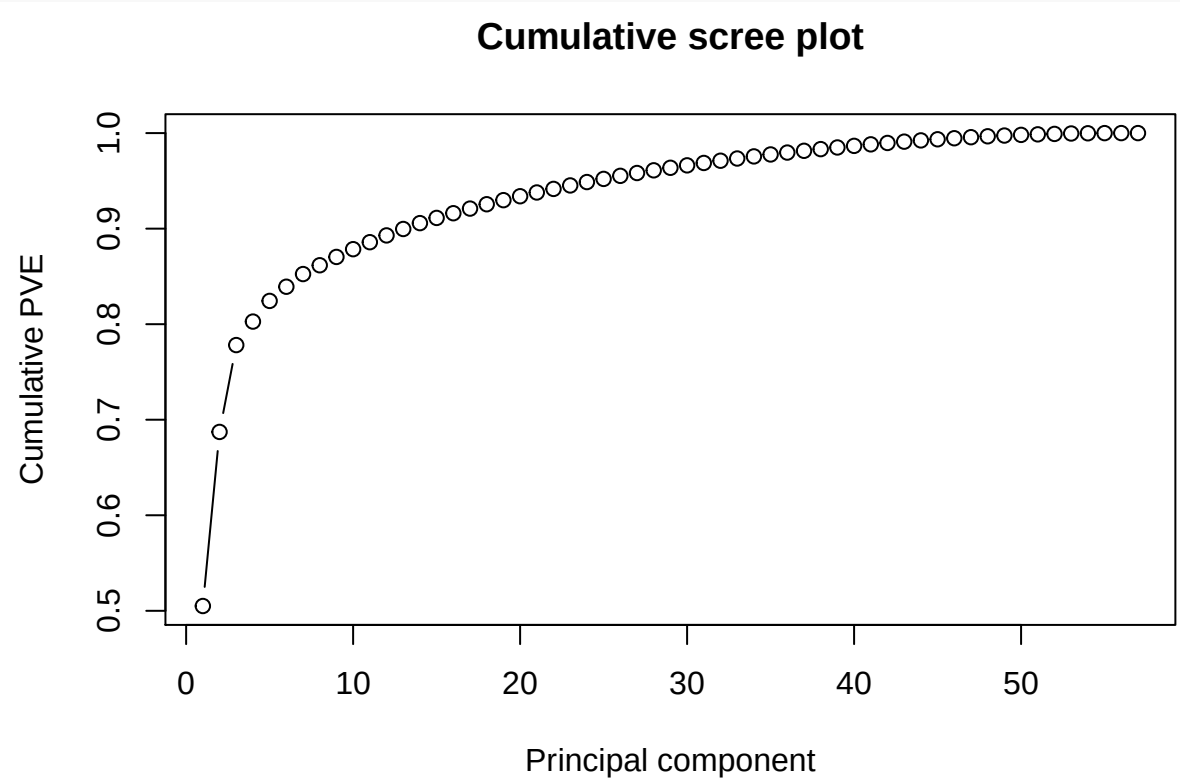


2.

```
# plot the variances against the number of the principal component
fit <- princomp(wiki_data)
screeplot(fit, npcs = 57, type = "lines")
```




```
# plot the cumulative PVE
cum_pve = cumsum(fit$sdev^2)/sum(fit$sdev^2)
plot(cum_pve, type='b', xlab='Principal component', ylab='Cumulative PVE', main='Cumulative scree plot')
```



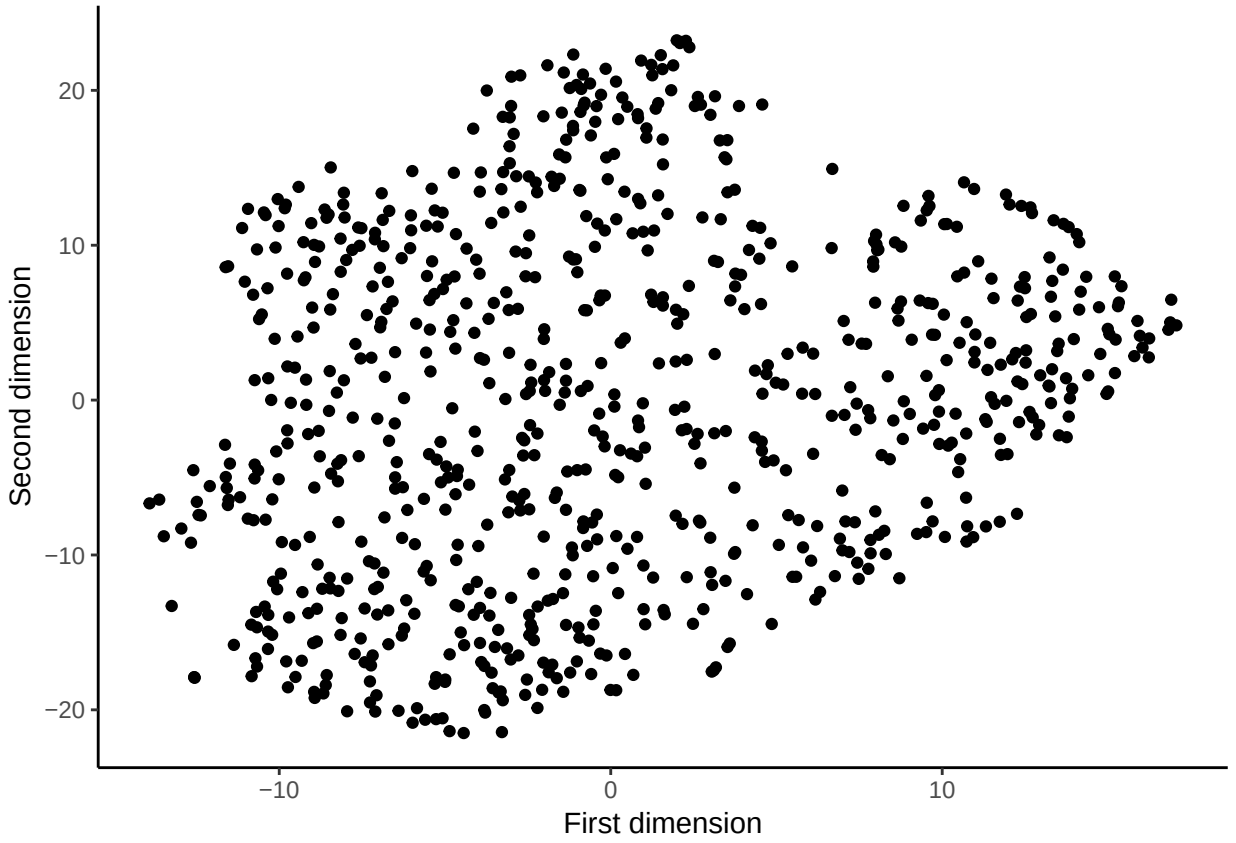

```
cum_pve[2]
```

```
##    Comp.2
## 0.6872382
```

About 68.7% of the variation is explained by the first two principal components.

3.

```
wiki_tsne <- Rtsne(wiki_data)

tsne1 = wiki_tsne$Y[,1]
tsne2 = wiki_tsne$Y[,2]

ggplot(wiki_data, aes(tsne1, tsne2)) +
geom_point() +
labs(x = "First dimension",
     y = "Second dimension") + theme_classic()
```



PCA uses a linear algorothm that is unable to interprest complex relationship between featrues well. On the contrary, t-SNE uses a non-linear algorithm so it has a stronger ability to interpret the complex polynomial relationship between features. t-SNE idenetifies observed clusters according to similarity. As the plots above, we can see that observations are more spread out in t-SNE plot.

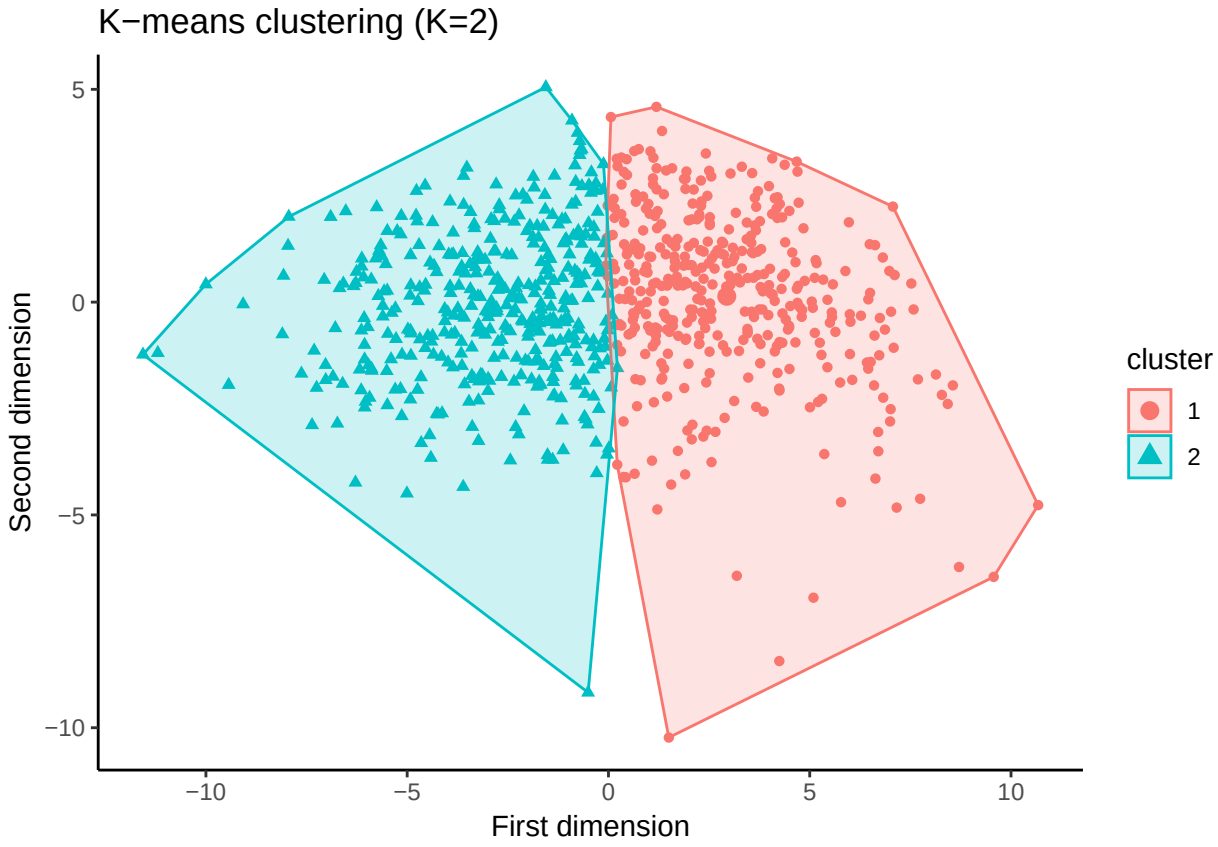
Clustering

1.

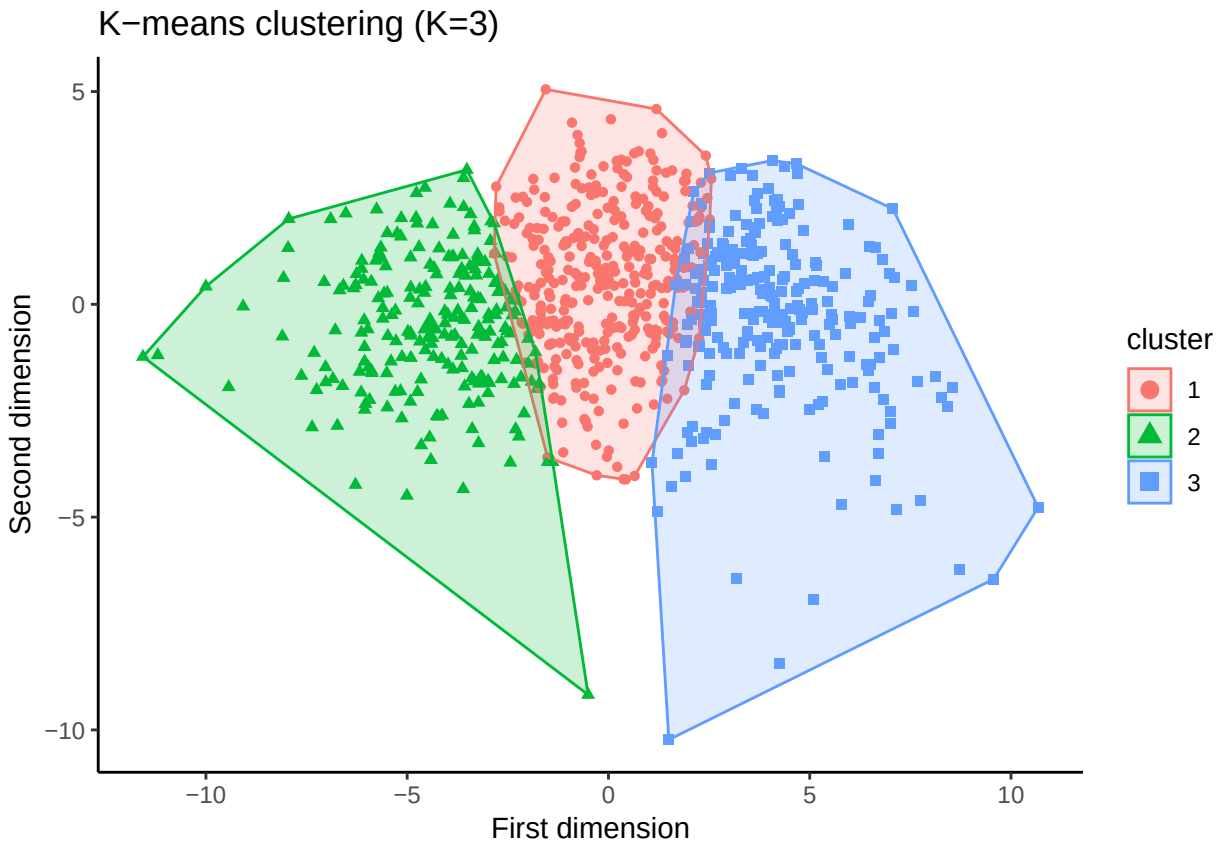
```
wiki_data_scaled <- scale(wiki_data)

wiki_k2 <- kmeans(wiki_data_scaled, centers = 2, nstart = 25)
```

```
wiki_k3 <- kmeans(wiki_data_scaled, centers = 3, nstart = 25)
wiki_k4 <- kmeans(wiki_data_scaled, centers = 4, nstart = 25)
```

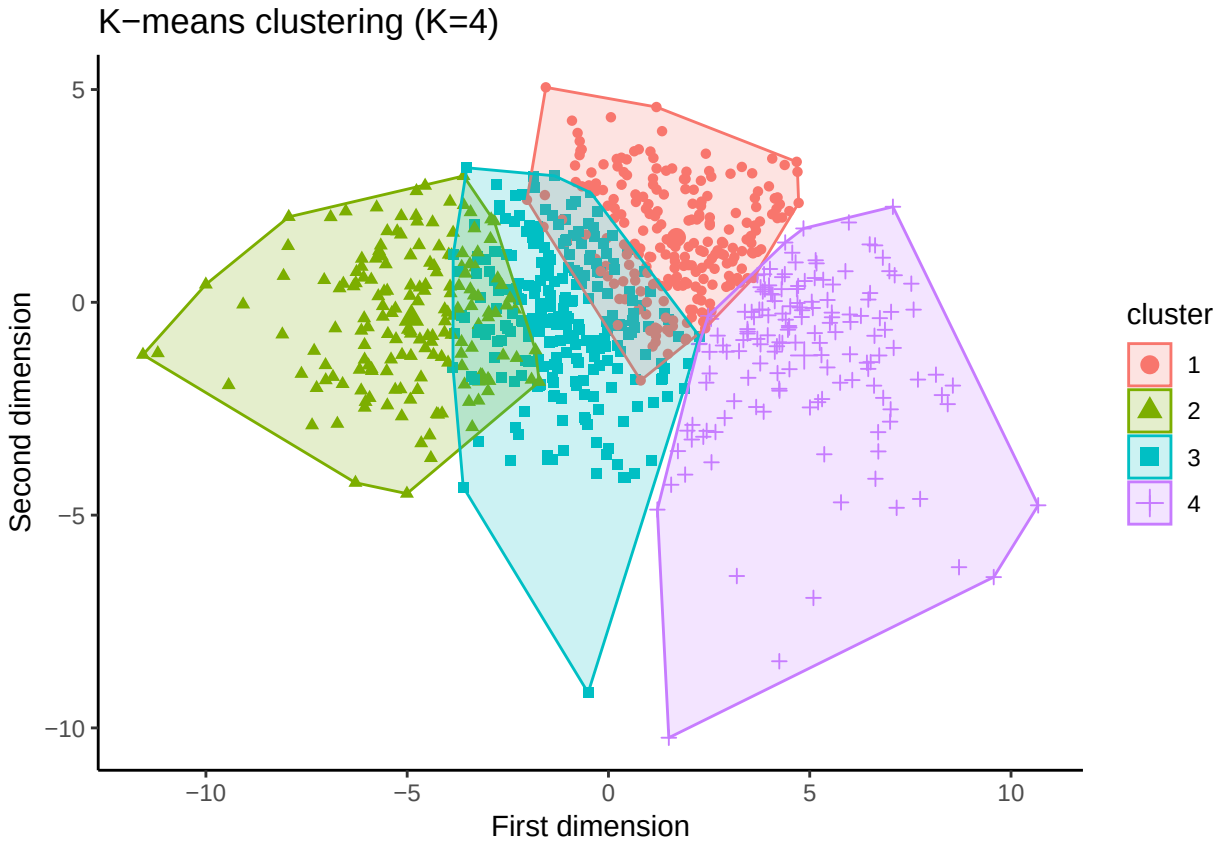
```
fviz_cluster(wiki_k2, data = wiki_data_scaled, geom = "point") +
  labs(title = "K-means clustering (K=2)",
       x = "First dimension",
       y = "Second dimension") + theme_classic()
```




```
fviz_cluster(wiki_k3, data = wiki_data_scaled, geom = "point") +
  labs(title = "K-means clustering (K=3)",
       x = "First dimension",
       y = "Second dimension") + theme_classic()
```



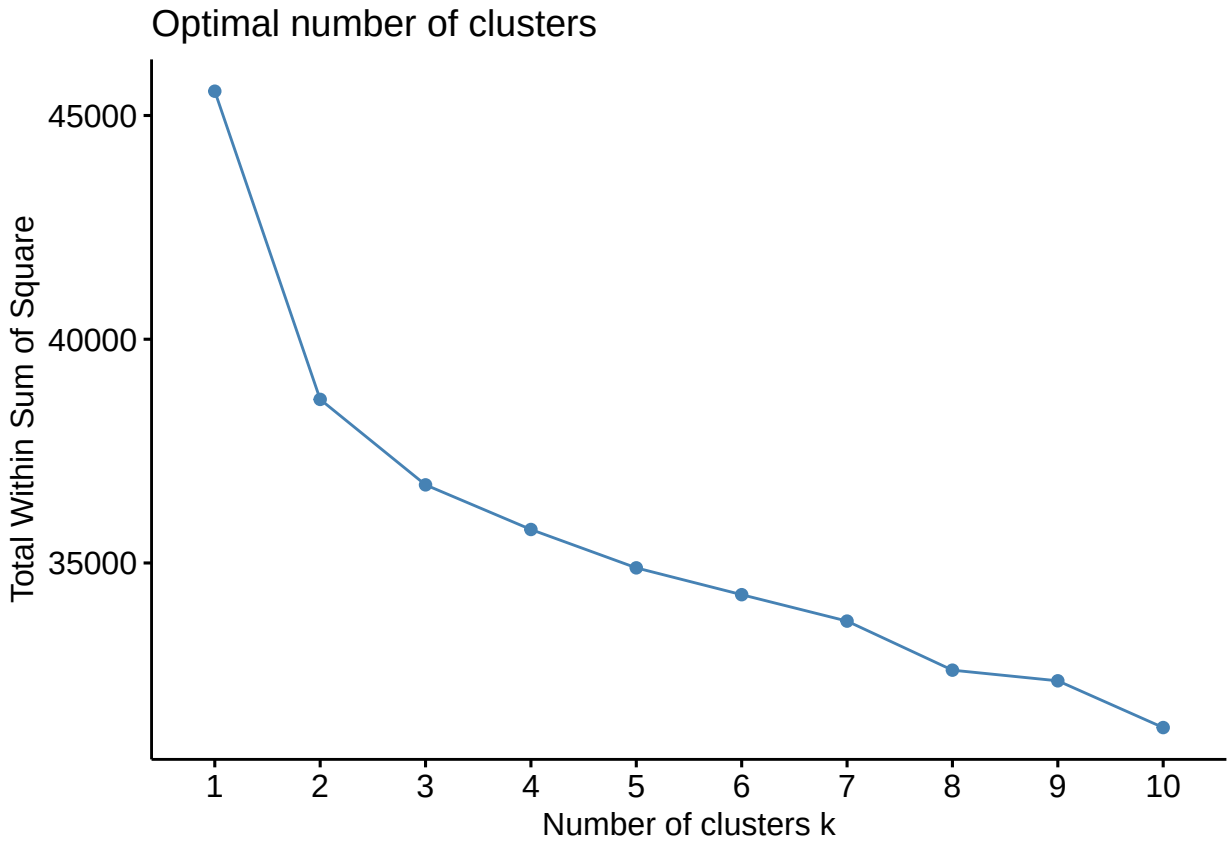

```
fviz_cluster(wiki_k4, data = wiki_data_scaled, geom = "point") +
  labs(title = "K-means clustering (K=4)",
       x = "First dimension",
       y = "Second dimension") + theme_classic()
```



According to the K-means clustering plots above, we can see that k=2 and k=3 provides similar outcome of clustering, which are pretty neat partitions. However, for k=4, the clusters seem to overlap with each other, which implies unclear partitions among the clusters.

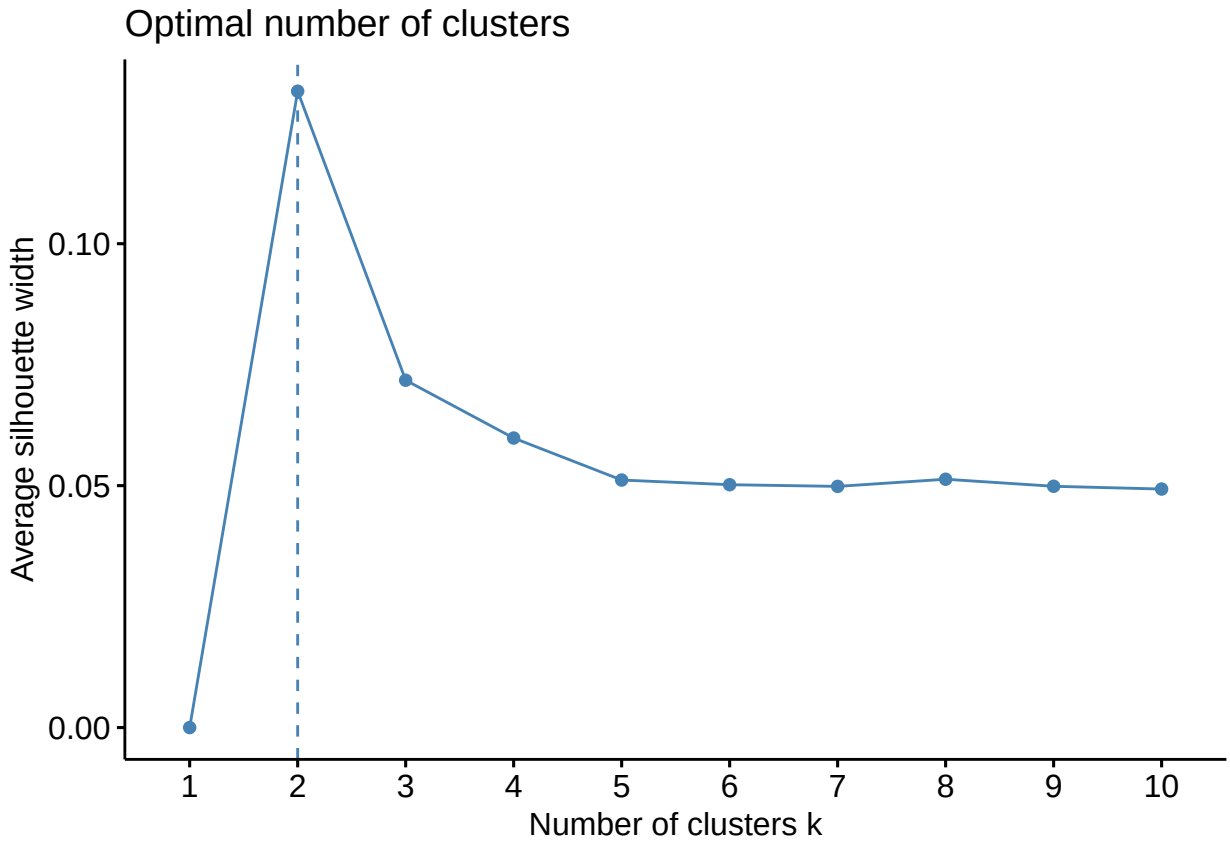
2.

```
# Elbow method
fviz_nbclust(wiki_data_scaled, kmeans, method = "wss")
```



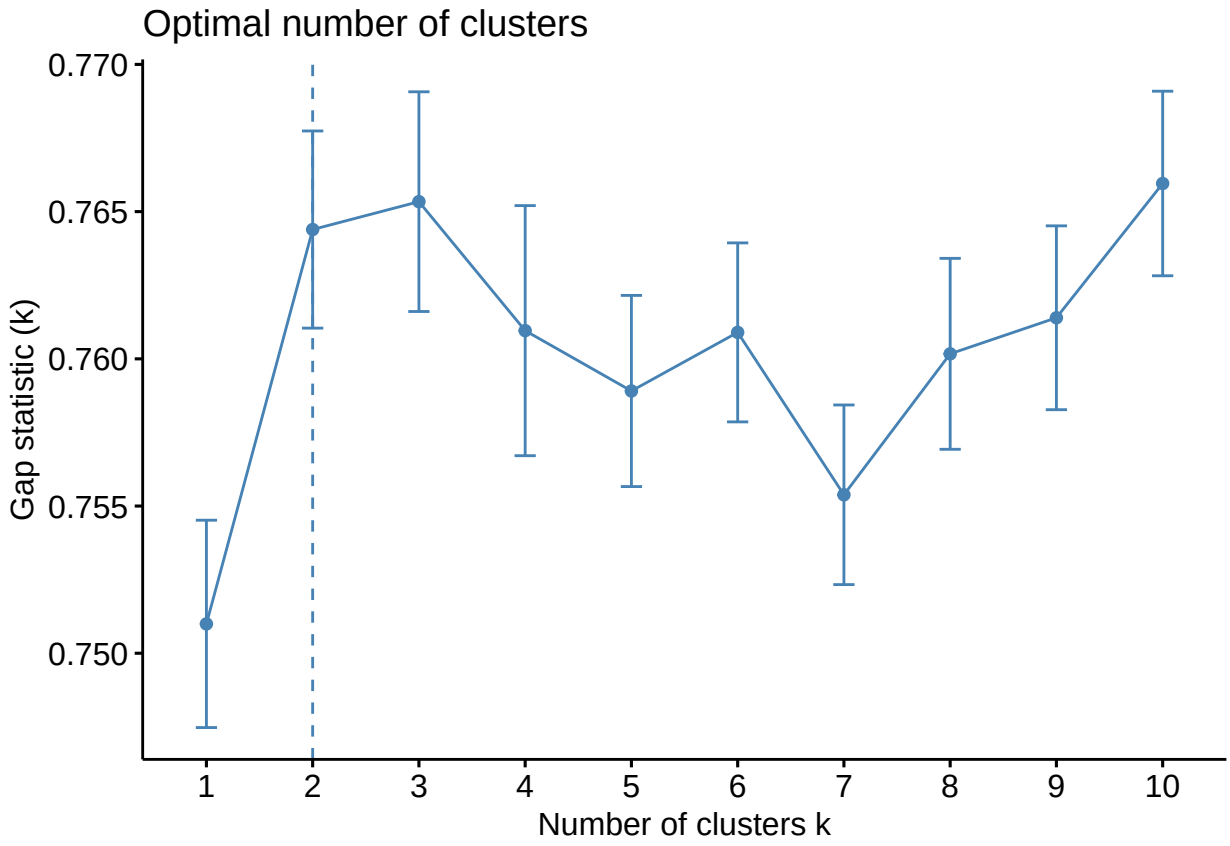
The total within sum of square drops dramatically when k=2 and slows down after k=3.

```
# Average silhouette
fviz_nbclust(wiki_data_scaled, kmeans, method = "silhouette")
```



The optimal number of clusters using average silhouette is 2.

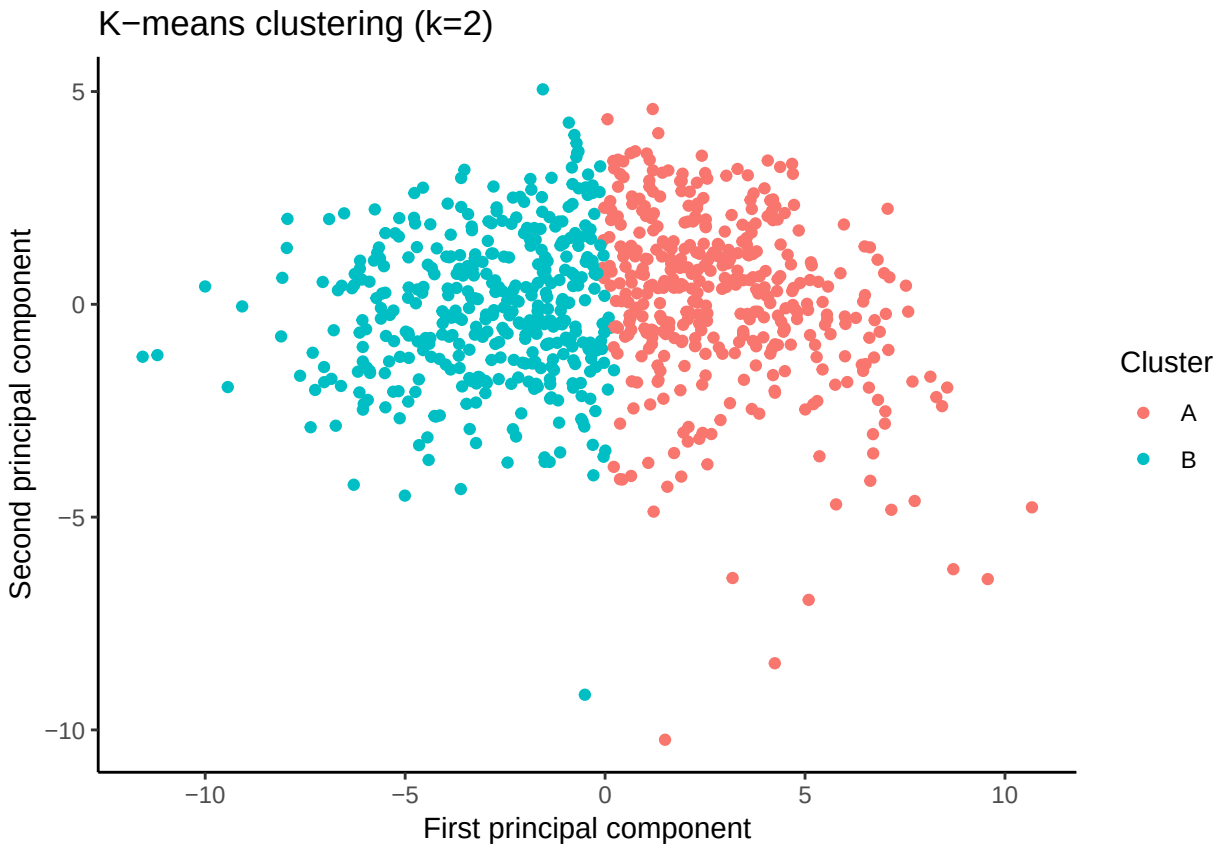
```
# Gap statistic
fviz_nbclust(wiki_data_scaled, kmeans, method = "gap_stat")
```



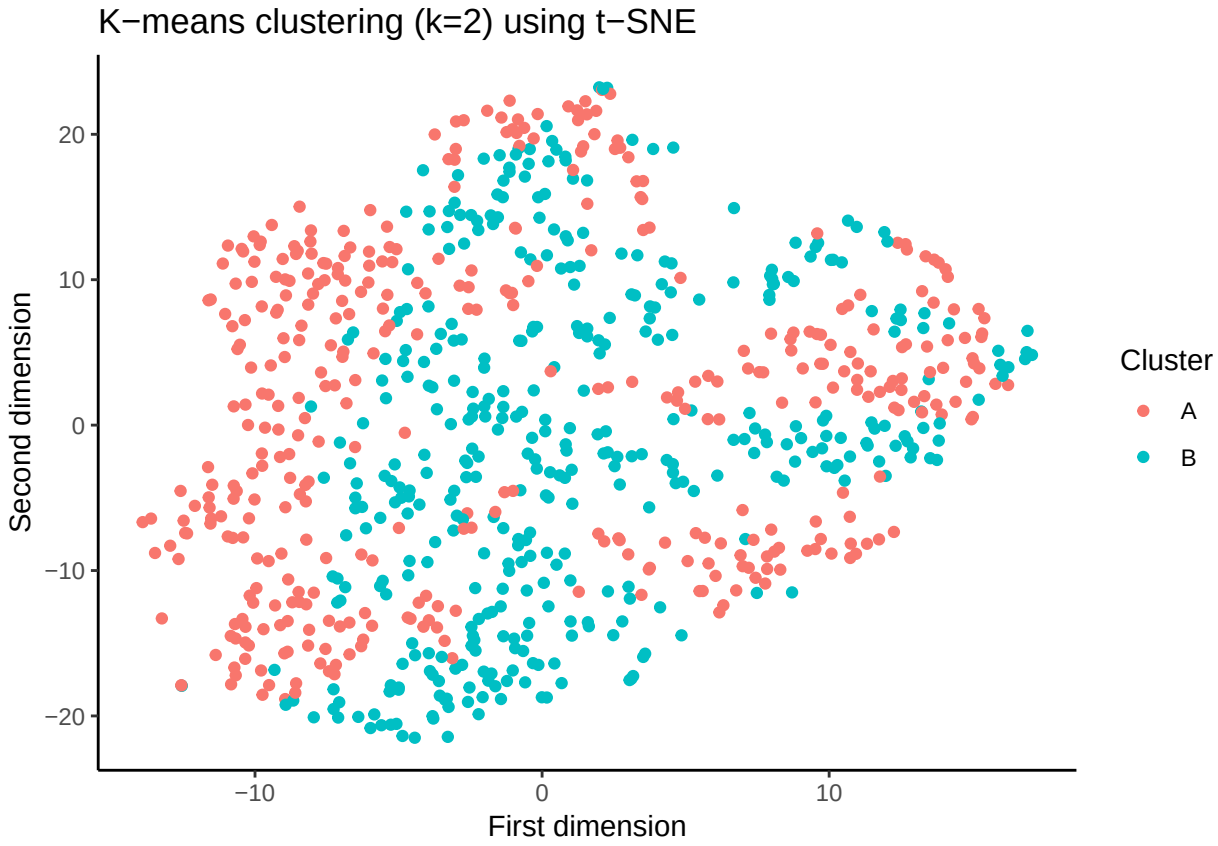
The optimal number of clusters using gap statistic is 2.

3.

```
wiki_data <- wiki_data %>%
  mutate(cluster = factor(wiki_k2$cluster, levels = 1:2, labels = c("A", "B")))

augment(wiki_pca, data = wiki_data) %>%
  ggplot(aes(.fittedPC1, .fittedPC2, color = cluster)) + geom_point() +
  labs(title = "K-means clustering (k=2)",
       x = "First principal component",
       y = "Second principal component",
       color = "Cluster") + theme_classic()
```




```
ggplot(wiki_data, aes(tsne1, tsne2, color = cluster)) + geom_point() +
labs(title = "K-means clustering (k=2) using t-SNE",
     x = "First dimension",
     y = "Second dimension",
     color = "Cluster") + theme_classic()
```



When reducing the dimensionality, PCA uses a linear approach and t-SNE uses a non-linear appraoch. While using the PCA approach, clusters are separated along positive or negative scale along the first principal component. On the otehr hand, using t-SNE will lead to a diagonal separation between clusters.

Exploring the clusters

1.

```
cv_ctrl <- trainControl(method = "cv", number = 10)

mars_grid <- expand.grid(degree = 1:3, nprune = seq(2, 25, length.out = 2) %>% floor())

tic()
set.seed(123)
mars_cv <- train(
  cluster ~ ., data = wiki_data, method = "earth",
  glm=list(family=binomial),
  trControl = cv_ctrl,
  tuneGrid = mars_grid,
  preProcess = c("zv")
)
```

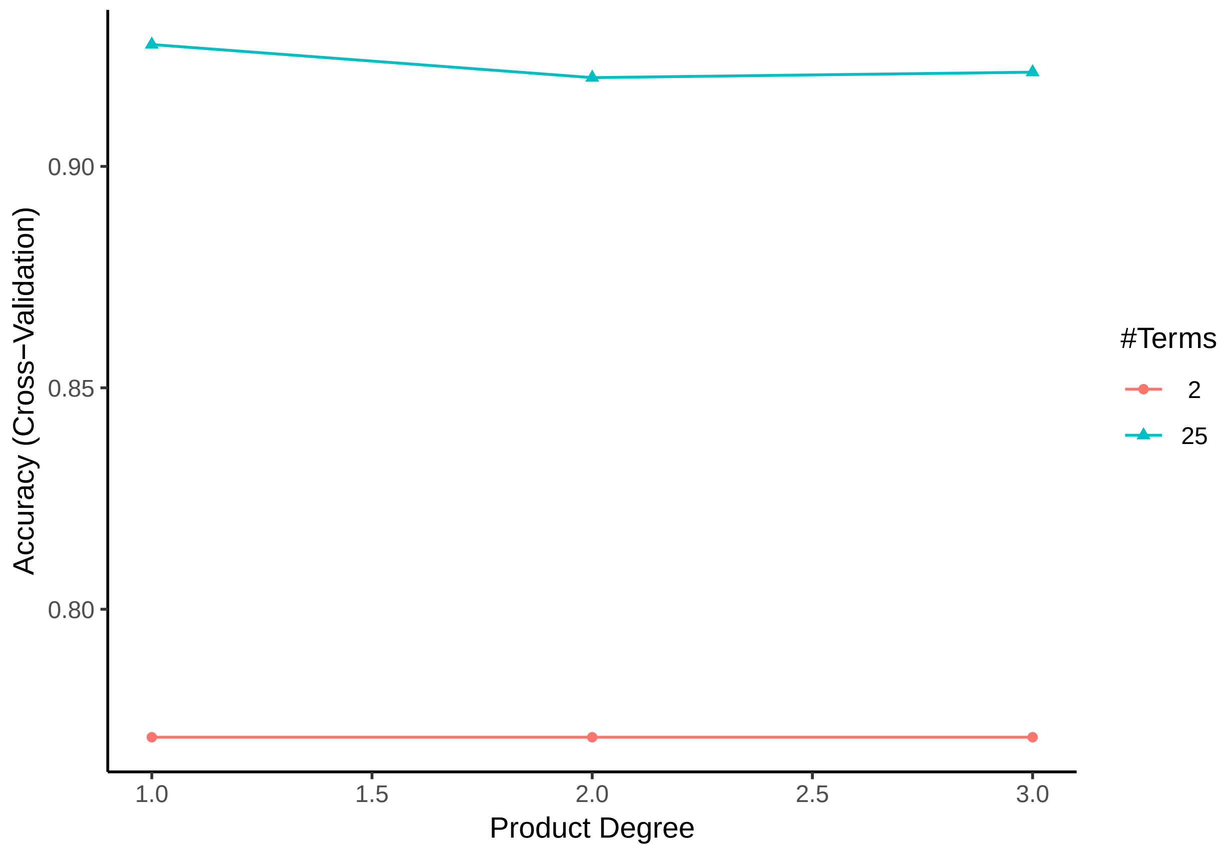
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
toc()
```

```
## 18.294 sec elapsed
```

```r
ggplot(mars_cv) + theme_classic()
```

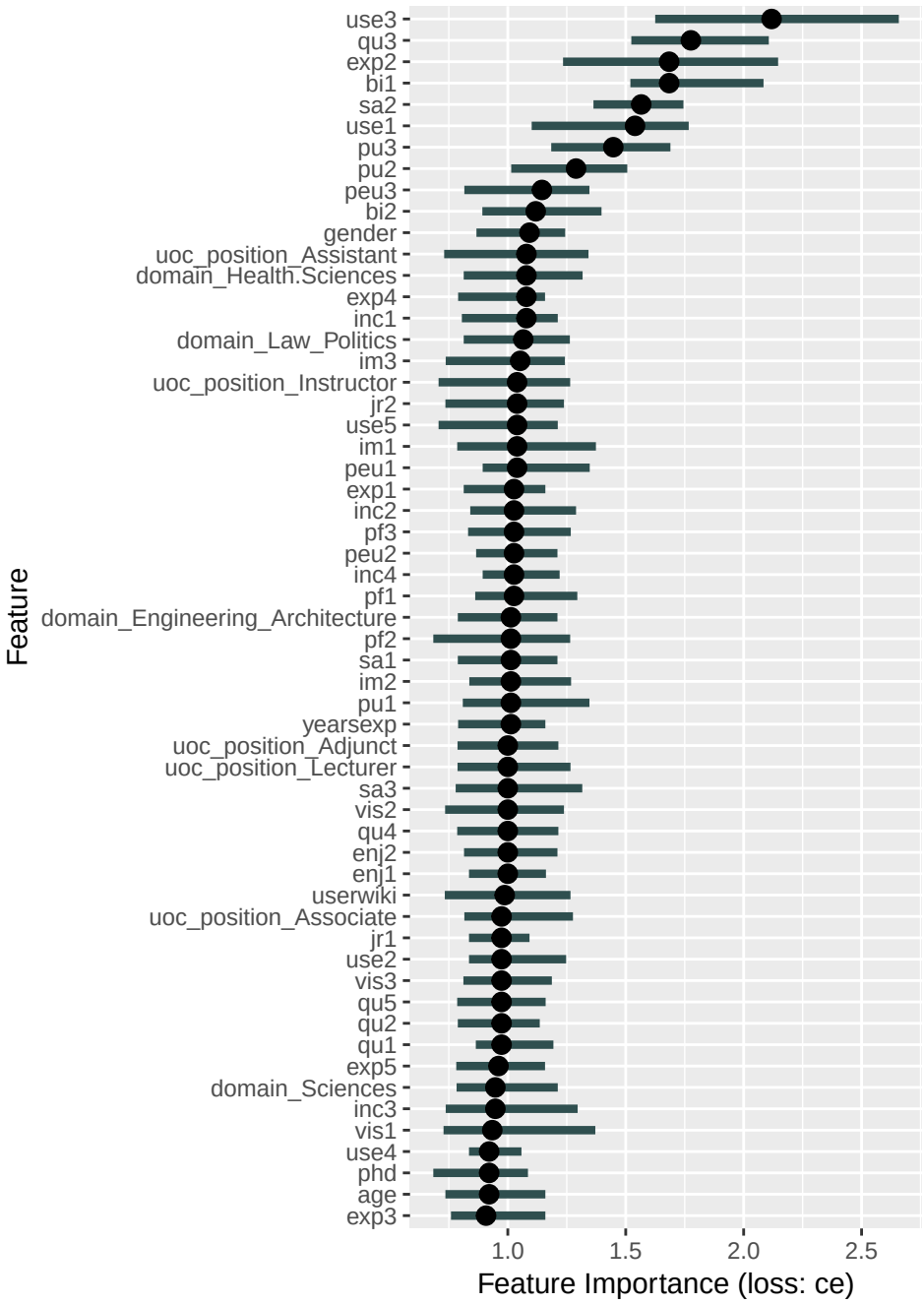


2.

```
data <- subset(wiki_data, select = -cluster)

pred_class <- Predictor$new(model = mars_cv, data = data,
                            y = wiki_data$cluster, type = "raw")
pred_prob <- Predictor$new(model = mars_cv,data = data,
                           y = wiki_data$cluster,class = "B", type = "prob")

mars_imp <- FeatureImp$new(pred_class, loss = "ce", parallel = TRUE, n.repetitions = 20)
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

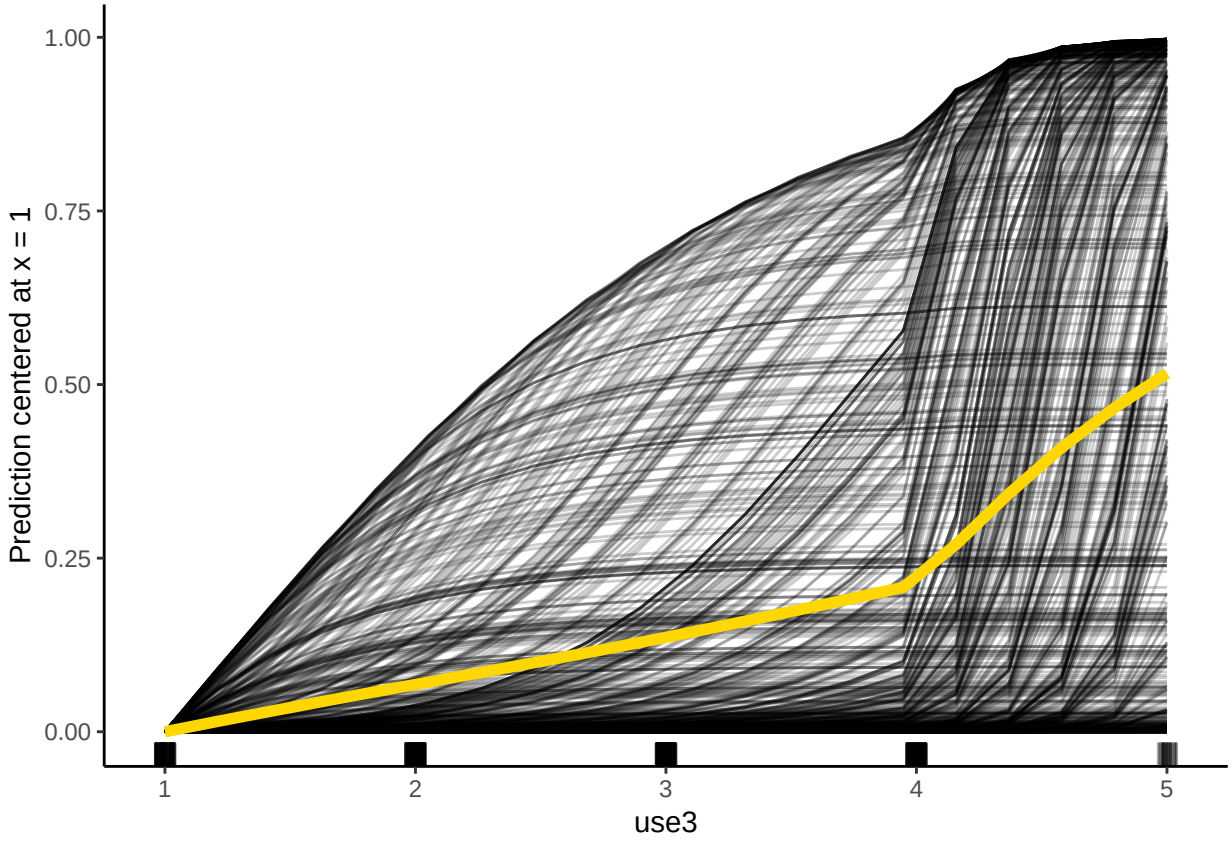
```
plot(mars_imp)
```



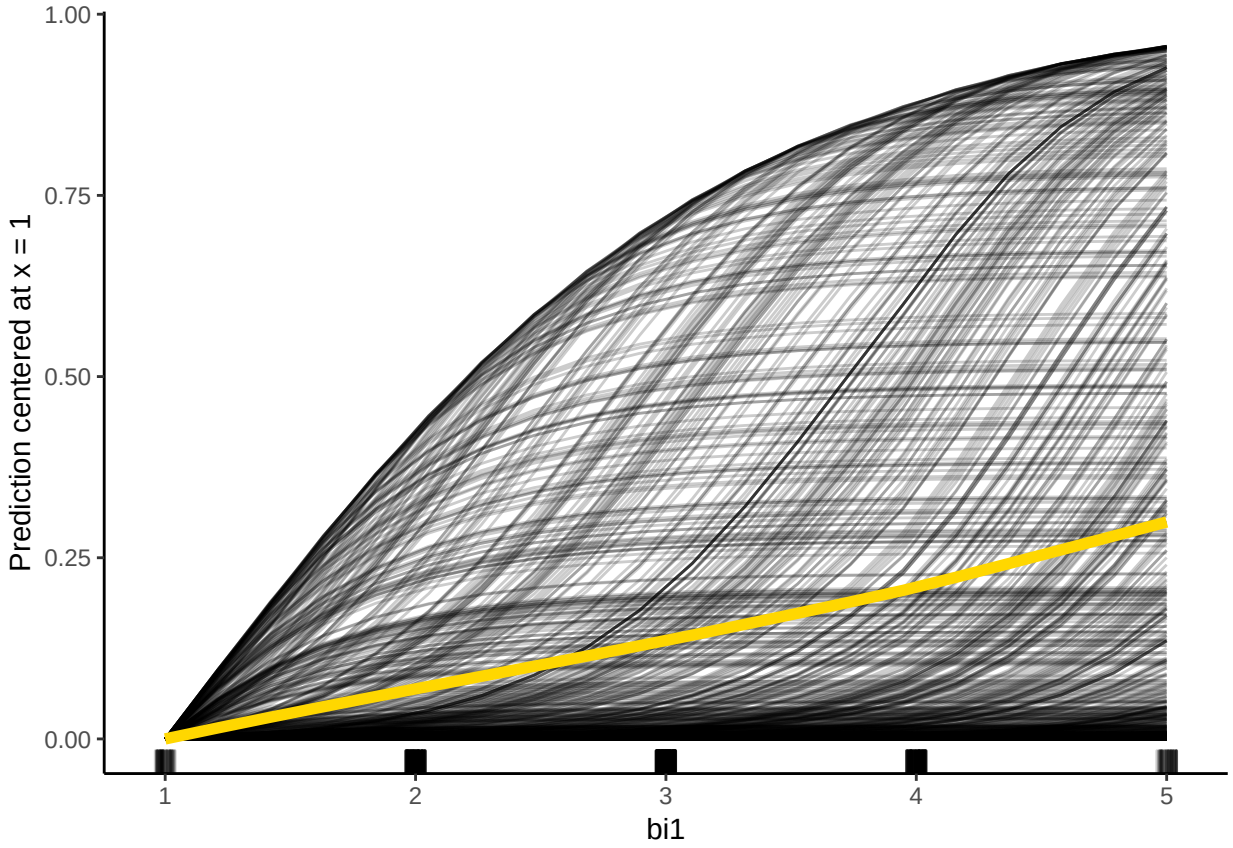
According to the plot above, we can know that the 5 most import features in wiki is 'use3', 'qu3', 'exp2', 'bi1', and 'as2'.

```
# using PDP/ICE to interprest the MARS
use3_pdp <- FeatureEffect$new(pred_prob, "use3", method = "pdp+ice",
                              center.at = min(wiki_data$use3))
bi1_pdp <- FeatureEffect$new(pred_prob, "bi1",method = "pdp+ice",
                             center.at = min(wiki_data$bi1))
qu3_pdp <- FeatureEffect$new(pred_prob, "qu3", method = "pdp+ice",
                             center.at = min(wiki_data$qu3))
exp2_pdpd <- FeatureEffect$new(pred_prob,"exp2", method = "pdp+ice",
                              center.at = min(wiki_data$exp2))
```

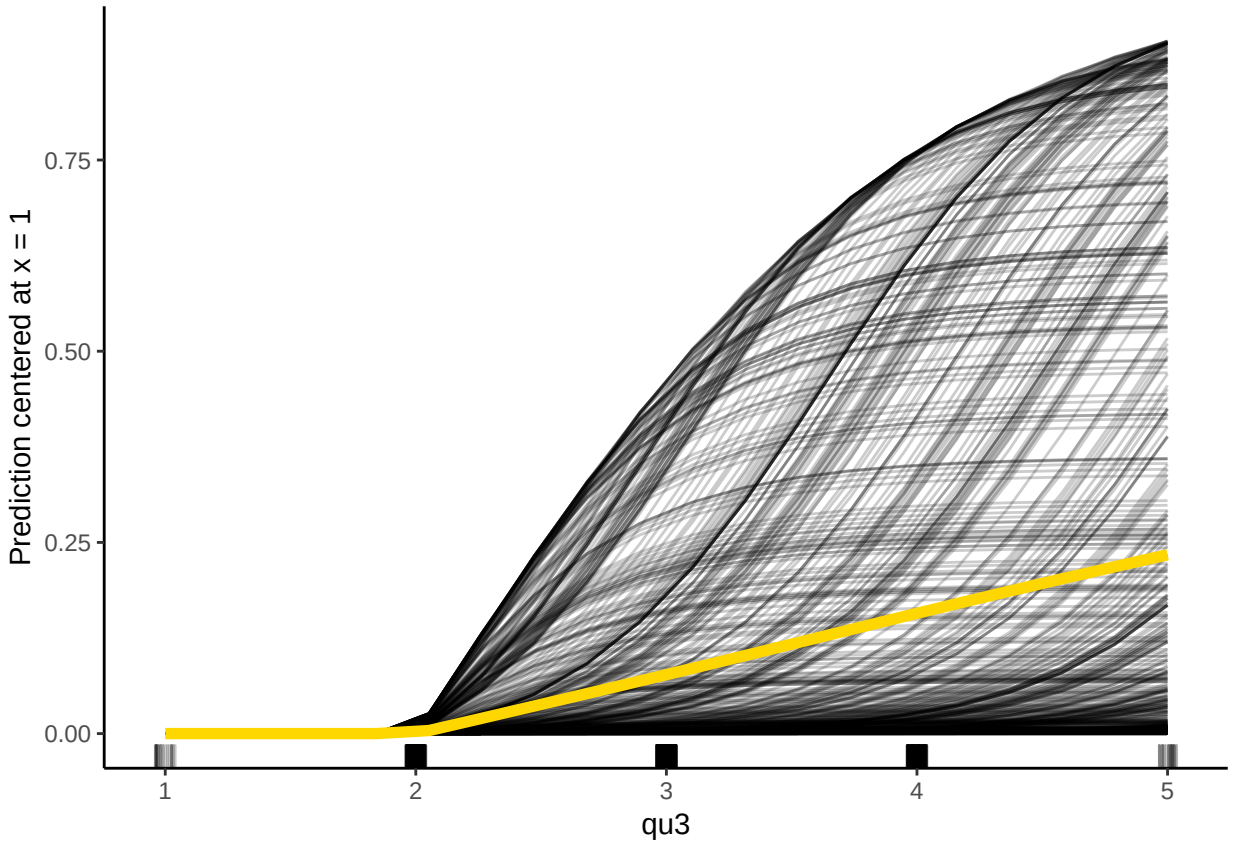
```
sa2_pdp <- FeatureEffect$new(pred_prob, "sa2", method = "pdp+ice",
                             center.at = min(wiki_data$sa2))

plot(use3_pdp) + theme_classic()
```



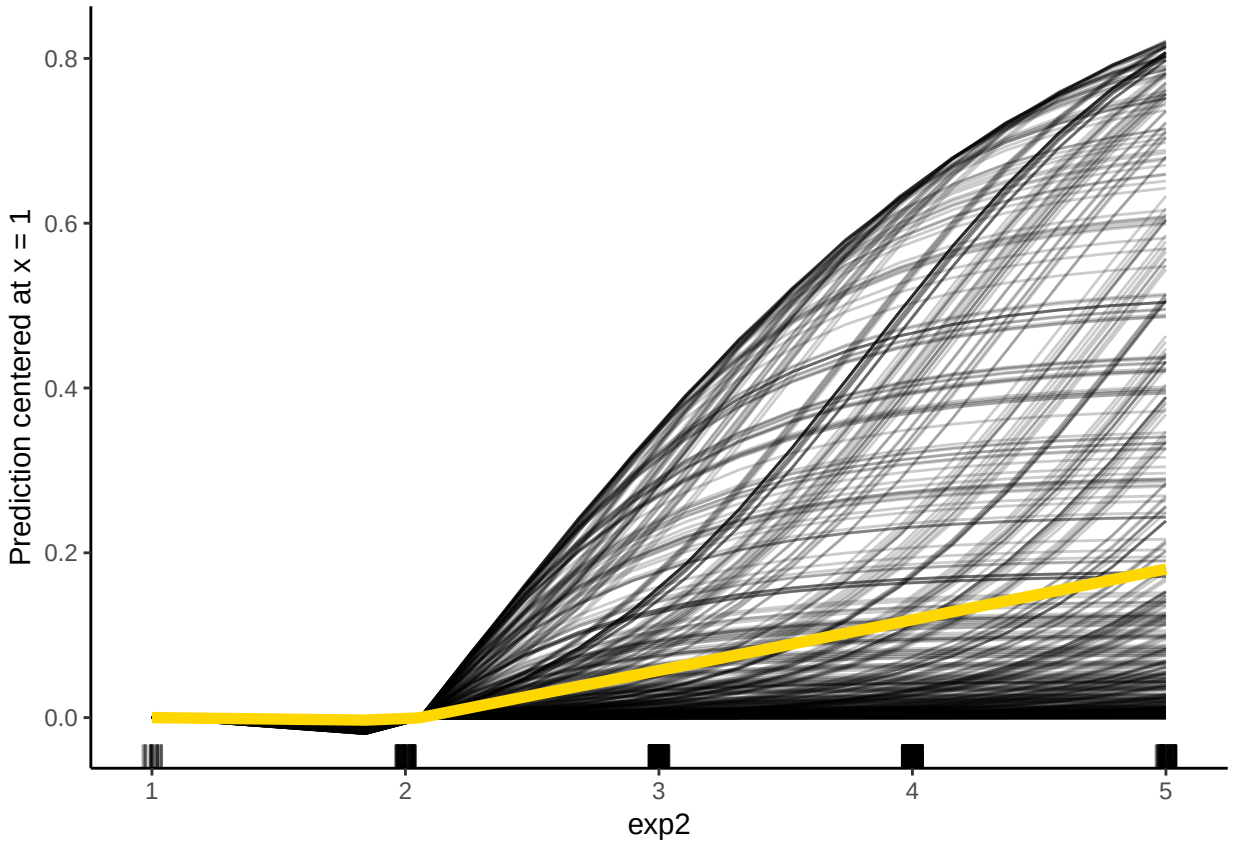

```
plot(bi1_pdp) + theme_classic()
```



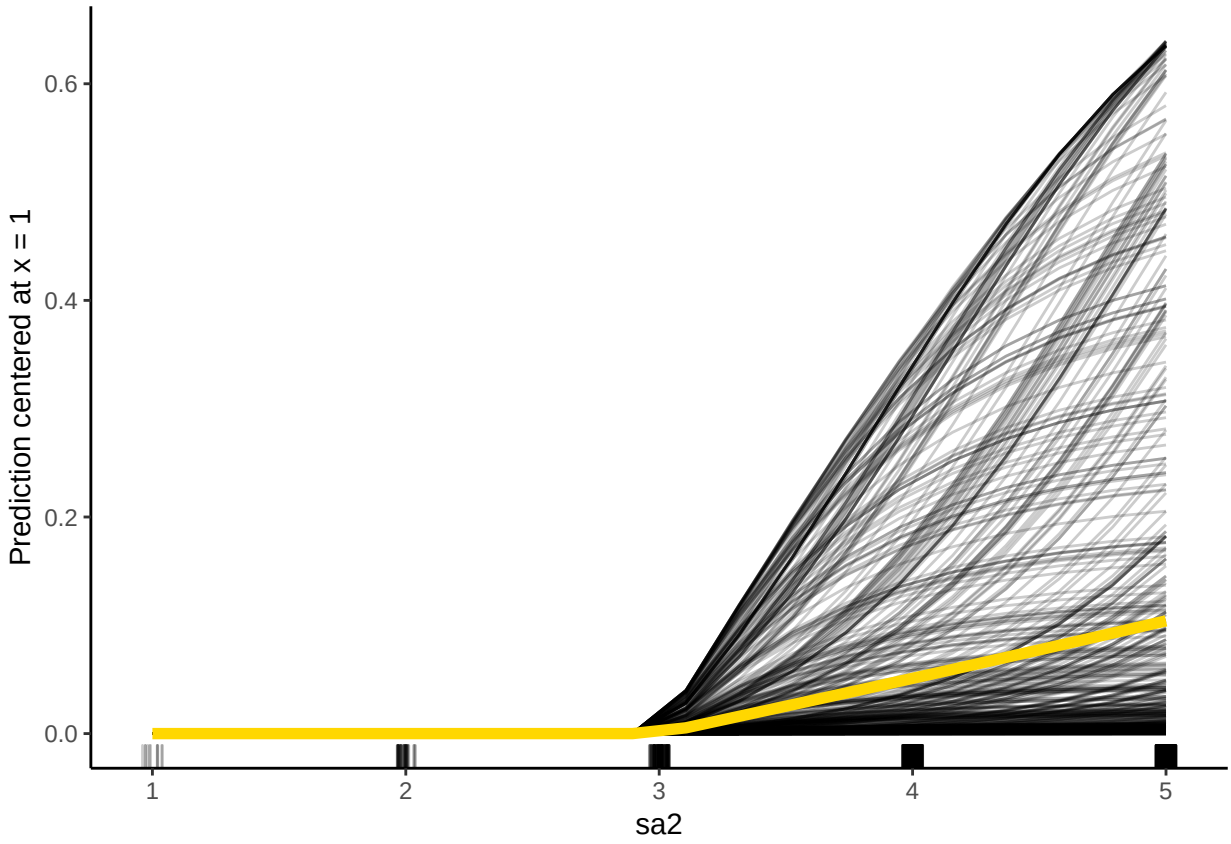

```
plot(qu3_pdp) + theme_classic()
```




```
plot(exp2_pdpd) + theme_classic()
```




```
plot(sa2_pdp) + theme_classic()
```



3.

The purpose of using clustering for Wikipediaa is likely to help identify supporters and opposers. The positive views of Wikipedia are realted to class 'B' versus class 'A'.